

1 Kathleen Sullivan (SBN 242261)
kathleensullivan@quinnemanuel.com
2 QUINN EMANUEL URQUHART &
SULLIVAN LLP
3 51 Madison Avenue, 22nd Floor
New York, NY 10010
4 Telephone: (212) 849-7000
Facsimile: (212) 849-7100

5 Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
6 John M. Neukom (SBN 275887)
johnneukom@quinnemanuel.com.
7 QUINN EMANUEL URQUHART &
SULLIVAN LLP
8 50 California Street, 22nd Floor
San Francisco, CA 94111
9 Telephone: (415) 875-6600
10 Facsimile: (415) 875-6700

11 Mark Tung (SBN 245782)
marktung@quinnemanuel.com
12 QUINN EMANUEL URQUHART &
SULLIVAN LLP
13 555 Twin Dolphin Drive, 5th Floor
Redwood Shores, CA 94065
14 Telephone: (650) 801-5000
Facsimile: (650) 801-5100

15
16 *Attorneys for Plaintiff Cisco Systems, Inc.*

Steven Cherny (*admitted pro hac vice*)
steven.cherny@kirkland.com
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, New York 10022
Telephone: (212) 446-4800
Facsimile: (212) 446-4900

Adam R. Alper (SBN 196834)
adam.alper@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, California 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

Michael W. De Vries (SBN 211001)
michael.devries@kirkland.com
KIRKLAND & ELLIS LLP
333 South Hope Street
Los Angeles, California 90071
Telephone: (213) 680-8400
Facsimile: (213) 680-8500

17
18 **UNITED STATES DISTRICT COURT**

19 **NORTHERN DISTRICT OF CALIFORNIA, SAN JOSE DIVISION**

20
21 CISCO SYSTEMS, INC.,

22 Plaintiff,

23 vs.

24 ARISTA NETWORKS, INC.,

25 Defendant.

CASE NO. 5:14-cv-5344-BLF

DECLARATION OF PHILLIP REMAKER

Date: September 29, 2015
Time: 1:30 P.M.
Dept.: Courtroom 5

DEMAND FOR JURY TRIAL

1 I, Phillip Remaker, declare as follows:

- 2 1. I am a Distinguished Engineer at Cisco Systems, Inc. ("Cisco"). Except as otherwise
3 stated below, I have personal knowledge of the facts stated in this declaration and could
4 testify competently thereto.
- 5 2. I have worked at Cisco since 1992. I have held the title of Distinguished Engineer since
6 2002. I have held other titles at Cisco including Technical Leader, Technical Support
7 Manager, Manager, and various Engineer titles. I have a proficient understanding of
8 Cisco's software document management and source code repositories, and knowledge
9 about the duties of certain current and former employees.
- 10 3. Cisco does not have uniform, centralized document management systems that store
11 documents relating to every product, release, or project that Cisco has developed or
12 commercialized. Cisco has numerous document management systems at every level of the
13 company structure, from company-wide to project-specific. To the best of my knowledge,
14 Cisco does not, as a matter of course, maintain a list or lists of all document management
15 systems, or the specific document management system that may contain documents related
16 to a specific product or project.
- 17 4. The largest document management system is called Engineering Document Control
18 System, or "EDCS."
- 19 5. EDCS stores certain documents related to current and past products, releases, and projects.
20 To the best of my knowledge, Cisco does not as a matter of course, maintain a list or lists
21 of the products, releases, and projects, or the types of document for each product, release,
22 and project, stored in EDCS. There are common types of documents that many, but not
23 all, products, releases, and projects create, utilize, and store in EDCS. These document
24 types include Project Requirements Documents and System Functional Specifications.

- 1 6. There are two ways to search EDCS. First is called FAST, a basic full- text search that
2 allows users to perform a text search through contents of all documents in the database,
3 described further below. The search results are limited to the first 500 results and
4 presented to the user in either “relevance order” or “reverse chronological order” (newest
5 first). The user is unaware of search results beyond the first 500. The second search
6 functionality allows users to limit search the following specific fields: Content, Title,
7 Filename, Doc. No., Author ID, Doc Type, Technology, and wwwwin-eng.cisco.com,
8 Cisco’s internal website for newsletters, employee training and reference materials, and
9 high-level project information. Some of these fields are filled in by users and others are
10 automatically populated by EDCS. Because there are many independent groups that
11 develop complex documents using common document templates, the consistency of the
12 documents can vary. There is no entity that routinely ensures that each or any of these
13 fields are populated for particular documents or that there is uniformity in the format or
14 type of information in any specific field. Moreover, this search functionality only returns
15 exact matches. For example, the user must know the exact feature, product, code name or
16 document name to successfully search by document name. Searching by more than one of
17 these criteria requires manually crafting a search query in the specific database language
18 utilized by EDCS. This requires intimate knowledge of the database language, database
19 structure, and data. To my knowledge, in order to use this functionality, Cisco would need
20 to identify engineers with knowledge about a specific command expression along with
21 several skilled engineers with specific, non-overlapping knowledge of the product line,
22 database structure, and query language.
- 23
24
25
26 7. For both types of search functionality, even if documentation was loaded and maintained
27 in EDCS, simply searching for the exact text of a command expression is likely to return a
28

1 very large number of results since many documents may contain that command or words
2 contained in the command. Determining which one of those results is related to the
3 earliest release of the command expression can only be done by manually reviewing each
4 document. There is no guarantee that documents in EDCS are the earliest documents
5 related to any command expression since the command expression may have predated the
6 document control system.

7
8 8. Cisco's internal network (intranet) includes "crawling" (or "spidering") search engines
9 called TOPIC and FAST, which build a full-text index of a subset of several databases,
10 including issue tracking (DDTS/CDETS), Service Request (case) tracking, and some
11 recent mailing list archives. Although mailing list archives are not automatically saved,
12 certain mailing lists took initial steps to enable this functionality. Very few mailing lists
13 elect this option. TOPIC and FAST allow a user to conduct a high-level keyword search of
14 those databases. TOPIC and FAST are implementations of third-party search engines that
15 extracts data from databases, but the user may not be able to refine the search based on
16 specific database fields. This type of search technology is most useful for complex search
17 terms, such as command expressions comprised of uncommon terms. For example,
18 searching for a command expression that contains the word "interface" would yield so
19 many results that it would not be useful.

20
21
22 9. Cisco does not have a single source code repository server. Like its document
23 management systems, certain project teams may be allowed to create and maintain source
24 code repositories in a manner that best fits their needs. As a result, Cisco teams may use
25 several different types of source code repository technologies, such as CVS, Subversion,
26 and ClearCase.

27
28 10. To the best of my knowledge, Cisco does not as a matter of course maintain a list or lists of

1 the location of the source code repository servers for each and every project, product,
2 release, or project. Thus, to locate a source code repository server for the first time that a
3 command expression was introduced, I (or anyone else) would need to know not only the
4 product and/or product code name and the specific release and/or version number in which
5 that command expression appeared, but also have sufficient institutional knowledge to
6 know which source code repository server contains the source code (name, server network
7 (IP) address). In some instances, the latter may only be obtained by conducting a series of
8 informal discussions with people who may be familiar with the specific product or release
9 and tracking down a person with the requisite knowledge.

10
11 11. As a matter of course, Cisco does not track the software developer responsible for each line
12 of source code. Cisco teams work collaboratively and the person who “checks-in” (or puts
13 the code into the source code repository) for the first time may or may not be the person
14 that designed the command expression. They may be, and in many instances, will likely
15 be, the person who performed the programming task after all creative design decisions
16 were already made.

17
18 12. I have been asked to assist outside counsel for Cisco to identify the people responsible for
19 various Command Line Interface (“CLI”) command expressions.

20
21 13. I reviewed a list of more than 500 command expressions. I have not exhaustively
22 investigated this list, but I do not expect the analysis in this Declaration to change if given
23 more time.

24
25 14. Certain of these command expressions appear to be command expressions that were
26 designed, developed, and released with a significant product release. For some of these
27 product releases, a design document may indicate why Cisco chose the original command
28 expressions introduced in that product release. This type of document is often stored and

1 can be found in EDCS.

2 15. Certain of these command expressions appear to be command expressions that were
3 designed, developed, and released as a part of an “incremental” release. An incremental
4 release adds one or more features to the product. Each feature is associated with one, but
5 usually many, multi-word commands. For some of these incremental releases, a design
6 document may indicate why Cisco chose the original command expressions introduced in
7 that product release.
8

9 16. Certain of these command expressions appear to be command expressions that were
10 designed, developed, and released as real-time software patches or in response to specific
11 customer requests, or commands for which the origin is simply unclear. For some of these,
12 a design document or issue tracking report may indicate why Cisco chose the original
13 command expressions introduced in that product release.
14

15 17. I was asked to and did research some of the commands from 500 command expressions
16 mentioned in paragraph 13 to describe exemplary processes for identifying at least one
17 person associated with the original development of a command expression.

18 18. One example is the “Ip rip v2-broadcast” command. I was able to determine the relevant
19 information in a fairly straight-forward manner due to my personal knowledge. I know
20 that this command expression was first described in the “RIP v. 2 Specification.” I
21 searched EDCS for “RIP v2” using the Title criteria, which returned no search results. I
22 performed a full-text search for “RIP v2,” which returned 23 documents. I went through
23 each document but the system functional specification (or document(s) with similar
24 information) was not among the search results. Next, I searched for “RIP v2” on TOPIC
25 and reviewed each of the search results. One of the search results included a software
26 patch report describing this command expression. This report identified a programmer
27
28

1 who worked on this command expression. This programmer may know about the origin of
2 the command expression, or can direct further investigation.

3 19. Another example is the "Ip tacacs source-interface" command. Based on my personal
4 experience, I am aware that this command expression was likely implemented as an
5 incremental change to a released feature, TACACS. TACACS is the Terminal Access
6 Controller Access Control System which conducts user authorization checks. I searched
7 for the command expression text using TOPIC, which returned numerous search results.
8 One of the results was a software patch report with the words "tacacs source-interface" in
9 the title. From the contents of this software patch report, I was able to determine that the
10 command expression was introduced in either TACACS v. 10.3 or v. 11.0. Based on my
11 personal experience, I was able to locate the source code repositories for these versions of
12 TACACS. I searched for the command expression in the version 10.3 source code
13 repository, which returned no results. I then search for the command expression in the
14 version 11.0 source code repository, which returned a search hit. I was able to determine
15 that this file had a source code check-in log and examined it to determine the programmer
16 responsible for checking-in the command expression. The check-in log, by definition,
17 does not separately identify the person(s) responsible for the design of the command
18 expression.
19
20
21

22 20. The last example is the "ip extcommunity-list expanded" command. I am not familiar with
23 this command so I first researched the command expression using Google. I searched for
24 the command expression in EDCS through a full-text search and criteria based search,
25 neither of which yielded no search results. In TOPIC, the search for this command
26 expression returned many results that described using used the command, but offered little
27 clue about its origin. I searched TOPIC for "excommunity-list" and sorted by
28

1 chronological order (oldest first). I found a document that included the command
2 expression and the email address of a Cisco employee that may have knowledge of this
3 command expression, but no definitive information. These results provide some leads as
4 to those who may have more information about the origin of this command expression.
5

6 21. As mentioned above, a programmer who checks-in the source code may not be the
7 person(s) who designed or originated the command expression.

8 22. The above are broad categorizations. The only way to determine the most efficient manner
9 of determining the specific people responsible for specific command expressions is to go
10 through the process described above in paragraphs 18 to 20 in a serial and systematic
11 manner. For any given command expression, I would need to determine if the general
12 description in paragraphs 14 and 16 apply by actually performing the exemplary steps
13 described in paragraph 18. If that process is unsuccessful, I would attempt a process
14 similar to that in paragraph 19. If that process is unsuccessful, I would attempt a process
15 similar to that in paragraph 20.
16

17
18 I declare under penalty of perjury that to the best of my knowledge, the foregoing is true and
19 correct.

20 Executed on this 23 day of September, 2015 in San Jose,
21 California:
22

23
24 
25

26 Philip Remaker
27
28